



## Seuran toimintaa

# Tietojenkäsittelytieteiden opetuksen kehittämisseminaari

Ari Korhonen  
Teknillinen korkeakoulu  
Tietotekniikan laitos  
Ari.Korhonen@tkk.fi

### Tiivistelmä

Ohjelmoinnin perusopetuksen verkosto järjesti yhteistyössä Tietojenkäsittelytieteen Seuran ja Teknillisen korkeakoulun kanssa kaksipäiväisen seminaarin, jossa pohdittiin muun muassa tietojenkäsittelytieteiden opetuksen lähtökohtia ja vaatimuksia. Pääpaino oli kuitenkin yhteistyön ja verkoston rakentamisessa yhteisten työ- ja opetusvälineiden esittelyjen kautta. Verkosto on samannimisen verkostohankkeen (Suomen virtuaaliyliopiston rahoittama kolmivuotinen hanke) aikana muotoutunut yhteisö. Samalla tilaisuus oli verkostohankkeen loppuseminaari. Tässä kirjoituksessa esitellään Ohjelmoinnin perusopetuksen verkosto ja Tietojenkäsittelytieteiden opetuksen SIG sekä valotetaan seminaarin sisältöjä ja työskentelymuotoja. Samalla pohditaan myös verkoston tulevaisuutta ja erityisesti sen merkitystä Seuran piirissä toimivan SIGin kannalta.

## 1 Johdanto

Ohjelmoinnin perusopetuksen verkosto on alun perin samannimisen verkostohankkeen ympärille muodostunut yhteisö. Verkostohanke on saanut hankerahoitusta Suomen virtuaaliyliopistolta vuosiksi 2006–2008. Tämän rahoituksen turvin on käynnistetty muun muassa lukuisia pilot-tihankkeita, joissa on pyritty levittämään opetuksen tueksi kehitettyjä välineitä kotimaisten yliopistojen kesken. Lisäksi on järjestetty seminaareja, joiden työpajoissa välineiden käyttöä on esitelty ja koulutettu. Teknillisessä korkeakoulussa lokakuun lopulla järjestettiin hankkeen loppusemi-

naari. Aikaisemmat seminaarit on järjestetty Turussa vuonna 2006 sekä Tampereella ja Jyväskylässä keväällä ja syksyllä 2007. Seminaareihin on osallistunut joka kerta noin 40–50 tietojenkäsittelytieteiden opetuksesta vastaavaa opettajaa tai välineiden kehityksessä mukana olevaa tutkijaa. Verkosto toimii osittain päällekkäin ja järjestää seminaareja yhteistyössä Tietojenkäsittelytieteen Seuran sisällä toimivan Tietojenkäsittelytieteiden opetuksen SIGin kanssa.

Tässä kirjoituksessa kerrotaan aluksi, luvuissa 2 ja 3, yleisesti Ohjelmoinnin perusopetuksen verkostosta sekä Tietojenkäsittelytieteiden opetuksen SIGistä. Sen

jälkeen, luvussa 4, on raportti verkostohankkeen loppuseminaarista. Lopuksi, luvussa 5, on lyhyt yhteenveto, jossa on muun muassa lisätietoa verkoston verkkosivuista, postituslistoista ja niin edelleen.

## 2 Ohjelmoinnin perusopetuksen verkosto

### 2.1 Taustaa

Ohjelmoinnin perusopetus on kaiken tietotekniikan opiskelun perusta. Perusymmärrys ohjelmointikielistä, ohjelmointiprosessista, erilaisista tietorakenteista ja algoritmeista ja näihin kaikkiin liittyvistä abstraktioista on välttämätön pohja, jotta voisi osallistua menestyksellisesti tietoteknisten järjestelmien suunnitteluun ja toteuttamiseen. Tämä pohja saavutetaan tyypillisesti yliopisto-opinnoissa kahden ensimmäisen opiskeluvuoden aikana. Kurseja suorittavat tietotekniikan pääaineopiskelijoiden lisäksi myös muiden laitosten sivuaineopiskelijat. Esimerkiksi TKK:lla tyypillinen määrä tietotekniikkaa kaikilla koulun opiskelijoilla on noin 20 opintopistettä sisältäen työvälitkurssin, kaksi suurta ohjelmointikurssia sekä Tietorakenteet ja algoritmit -kurssin. Pääaineopiskelijoilla sekä eräillä aloilla, joissa tietotekniikka on erityisen tärkeässä roolissa, kurssien ja opintopisteiden määrä on luonnollisesti suurempi.

Ohjelmoinnin perusopetuksen verkosto on yksi monista matemaattis-luonnontieteellisellä alalla toimivista yliopistoverkostoista. Yliopistojen tieteenalojen verkostot rakentavat jonkin tieteenalan tai monitieteistä valtakunnallista toimintaverkkoa. Näistä verkostoista ja niiden kotisivuista pitää listaa Suomen virtuaaliyliopisto<sup>1</sup>. Verkostojen sivuilta löytyy tietoa

muun muassa alan kursseista, tutkijakouluista ja ajankohtaisista tapahtumista.

### 2.2 Motivaatio

Ohjelmoinnin peruskursseilla koulutus on valtakunnallisesti laajaa, ja yksittäisessä oppilaitoksessa tyypillisesti kaikille näille kursseille osallistuu satoja opiskelijoita. Johtuen suurista opiskelijamääristä aiheen opettamisen ja oppimisen tueksi on kehitetty erilaisia ohjelmistoja, joita voidaan käyttää muun muassa antamaan automaattisesti palautetta opiskelijoille ja helpottamaan harjoitustehtävien arviointia.

Automaattiseen tarkastamiseen kykenevien järjestelmien kehittäminen vie tyypillisesti useita henkilötyövuosia. Vastineeksi niiden edut erityisesti suurten opiskelijaryhmien opetuksessa voivat olla hyvin merkittäviä. Niiden avulla voidaan opiskelijoille tarjota paljon enemmän palautetta harjoitustehtävistä kuin mitä pelkillä henkilöresursseilla olisi mahdollista. Ohjelmoinnissa tämä on erityisen tärkeää, koska asiaa ei käytännössä voi oppia pelkästään lukemalla kirjallisuutta, vaan käytännön ohjelmointiharjoitukset ovat ratkaisevan tärkeitä oppimisen kannalta. Automaattisen tarkastamisen turvin opiskelijoilla voidaan teettää riittävä määrä harjoitustehtäviä, joista ensiksikin voidaan antaa opiskelijalle palautetta, ja toiseksi ne voidaan tarkastaa, jolloin tuloksia voidaan hyödyntää arvioinnissa. Palautteen saamisen harjoituksista on oleellista, jotta riittävä motivaatio niiden tekemiseen säilyy ja opiskelija pystyy itsenäisesti saamansa palautteen turvin ohjaamaan omaa oppimisprosessiaan. Lisäksi automaattinen arviointi mahdollistaa opettajien työpanoksen suuntaamisen arvioinnista muunlaiseen opetukseen, koska ainakin osa tärkeästä tarkastustyöstä voidaan automatisoida.

<sup>1</sup>[http://www.virtuaaliyliopisto.fi/vy\\_verkostot\\_fin.asp](http://www.virtuaaliyliopisto.fi/vy_verkostot_fin.asp)

Automaattinen tarkastaminen tarjoaa lisäksi kolme olennaista etua ihmisiin nähden. Ensinnäkin automaattiseen tarkastamiseen kykenevät järjestelmät soveltuvat erinomaisesti itsenäisen verkko-opiskelun tueksi, koska ne tarjoavat automaattista palautetta opiskelijan työstä silloinkin, kun opettaja ei ole läsnä — edes verkon välityksellä. Näin ne tarjoavat opettajalle mahdollisuuden rakentaa monimuotoisia verkko-oppimateriaaleja tuomalla lisää vuorovaikutusta verkossa oleviin oppimisympäristöihin. Toiseksi opiskelijoille voidaan sallia tehtävän uudelleenpalautus automaattisen arvioinnin jälkeen jopa useaan otteeseen. Tällöin opiskelija voi oppia itsenäisesti omista virheistään analysoimalla saamaansa palautetta. Tyypillisesti aiheeseen liittyvät tehtävät ovat sen luonteisia, että oikeaa ratkaisua ei voi löytää yrityksen ja erehdyksen menetelmällä muutaman arvauksen jälkeen. Kolmas tärkeä etu on se, että osa tehtävistä voidaan räätälöidä täysin henkilökohtaisiksi, jolloin ratkaisujen kopiointi ei ole enää mahdollista. Näistä automaattisen tarkastamisen tuomista eduista on saatu erittäin hyviä tuloksia muun muassa TRAKLA2-järjestelmän kanssa [8].

Kaikki edellä mainitut edut on käytännössä todettu niissä oppilaitoksissa, joissa järjestelmiä on kehitetty ja käytetty. Vuosien käyttökokemukset ovat kuitenkin osoittaneet, että järjestelmien leviäminen oppilaitoksesta toiseen tapahtuu hyvin hitaasti ja sattumanvaraisesti. Tällöin järjestelmien kehittämiseen käytetty mittava työmäärä jää pääosin vain alkuperäisen välineen kehittäneen oppilaitoksen hyväksi. Lisäksi on havaittu, että usein eri oppilaitoksissa tehdään päällekkäistä työtä. Täysin toisiaan vastaavia järjestelmiä on kehitetty eri puolilla, joka haaskaa resursseja.

## 2.3 Tavoite

Ohjelmoinnin perusopetuksen verkoston tarkoituksena on levittää tietoa ohjelmoinnin opettamiseen liittyvistä parhaista käytännöistä ja aineistoista suomalaisissa yliopistoissa sekä edesauttaa tämän alueen opettajien verkostoitumista. Tavoitteena on vakiinnuttaa tasokas ja taloudellinen tieto- ja viestintätekniikan käyttöön vahvasti nojaava opetuksen ja tutkimuksen toimintatapa, joka hyödyntää myös viimeisintä alan tutkimustietoa.

Suomessa on tehty merkittävää, kansainvälisestikin noteerattua työtä ohjelmoinnin opetuksen liittyvien oppimisympäristöjen kehittämisessä. Monesti tulokset ja erityisesti toteutetut välineet ovat kuitenkin jääneet vain kehittäjäyhteisöjen omaan käyttöön. Verkoston tavoitteena onkin edesauttaa aikaansaatuun ratkaisujen leviämistä muihin oppilaitoksiin. Yhteistyöllä voidaan ohjelmoinnin oppimisen tasoa nostaa merkittävästi yhdistämällä voimia ja jakamalla kehitysresursseja sopivasti.

Ohjelmistojen levittämiseen tähtäävässä pilottihankkeissa on kuitenkin havaittu edellä mainittujen tavoitteiden toteutumisen esteinä muun muassa seuraavanlaisia asioita.

1. Automaattiseen tarkastamiseen perustuvat järjestelmät liittyvät lähes aina joihinkin tiettyihin harjoitustehtäviin. Vaikka ohjelmoinnin opiskelu on periaatteessa samanlaista joka puolella, opettajilla on kuitenkin aina omia painotuksiaan, millaisia tehtäviä he haluavat opiskelijoillaan teettää. Jos järjestelmän käyttöönotto edellyttää aiemmin toteutettujen tehtävien muokkaamista uusiin tarpeisiin tai kokonaan uusien tehtävien laatimista, useimmilla opettajilla ei ole täl-

laiseen aikaa. Tämä johtuu siitä, että automaattisissa välineissä edellytetään tyypillisesti ajankäyttöä välineen käyttöönottovaiheessa eli ennen kurssia, joka on ainakin “henkisesti työläin” vaihe. Hyöty tulee vasta myöhemmin kun tehtäviä ei tarvitse tarkastaa manuaalisesti. On helpompaa jatkaa vanhalla menetelmällä, jos opettajalla ei ole käytettävissään ylimääräisiä henkilöresursseja. Hyväkin järjestelmä saattaa siis helposti jäädä ottamatta käyttöön.

2. Järjestelmien yhteyteen tarvitaan välttämättä — tai ainakin siitä on merkittävää etua — muuta verkko-opetusmateriaalia, koska järjestelmiä voidaan vain harvoin sovittaa riippuvaiseksi esimerkiksi jostain tietystä oppikirjasta. Oppimateriaalin tuottaminen (teoria, selitykset, esimerkit, tehtävät, ...) ja muokkaaminen vaatii kuitenkin runsaasti työaika, ja harvoilla opettajilla on siihen mahdollisuutta virkatyötä hoitaessaan.
3. Itse järjestelmien kehittäminen rinnastuu tieteelliseen tutkimustyöhön, ja siitä voidaan kirjoittaa merkittävästi julkaisuja. Tähän voidaan saada myös ulkoista rahoitusta. Sen sijaan sisällön tuottaminen sekä teknologian ja hyvien käytänteiden levittäminen tai vastaanottaminen ei tuota minkäänlaista tieteellistä ansiota, ja tämän työn tueksi voidaan yleensä saada vain pienimuotoisia paikallisia apurahoja esimerkiksi korkeakouluilta. Tällaisen rahoituksen turvin ei kansallinen käytäntöjen levittäminen onnistu.

Suomen virtuaaliyliopistolta saadun hankerahoituksen turvin on pyritty edellä

kuvattujen esteiden purkamiseen. Rahoituksen turvin on useita työkaluja voitu kehittää, markkinoida ja ennen kaikkea tuoteistaa sellaiselle tasolle, että ne on voitu ottaa helposti käyttöön myös muissa kuin kehittäjäyliopistoissa. Esimerkkejä tällaisista järjestelmistä ovat jo aiemmin mainittu TRAKLA2 ja siihen integroidut ViLLE-tehtävät [10] ohjelmoinnin peruskursseille sekä Aloha [1], sen seuraaja Rubyrice [11] ja Formulaattori [2]. Kaikkia näitä välineitä voidaan käyttää joko yhdessä tai erikseen opetuksen tukena sekä automaattisen palautteen antamiseen. Yhteistyö välineiden kehittämisessä on ollut vilkasta (katso esimerkiksi [5, 6, 7, 9]). Yhteenvedon eräistä jopa tutkimusyhteistyöksi edenneistä verkottumisen muodoista voi lukea Tietojenkäsittelytieteen numerosta 23 [4]. Lisäesimerkkejä yhteistyöstä löytyy verkoston kotisivuilta löytyvistä pilottihankkeista, linkeistä sekä verkoston Wikistä.

Valitettavasti kaikkien välineiden leviäminen ei ole lähtenyt hyvin käyntiin. Asiaa on selvitetty kyselytutkimuksella, jonka tuloksia tullaan esittelemään myöhemmin julkaistavassa verkostohankkeen loppuraportissa. Sen alustavia tuloksia on esitelty luvussa 4.1.

## 2.4 Toiminta

Keskeisenä toimintamuotona ovat verkostotapaamiset ja seminaarit, joihin osallistuu alan opettajia, tutkijoita ja muutoin verkoston toiminnasta kiinnostuneita henkilöitä. Lisäksi verkostolla on oma tutkijaverkostonsa, joka järjestää pienimuotoisempia tapaamisia muun muassa jatko-opiskelijoille, joiden väitöskirjan aihe sivuaa opetusteknologiaa tai tietojenkäsittelytieteiden opetusta yleisemmin. Linkkinä verkoston ja (sen osajoukon) tutkijaverkoston jäsenten välillä toimii kaksi postituslistaa, joihin voivat liittyä vapaas-

Taulukko 1: Ohjelmoinnin perusopetuksen verkoston ja verkostohankkeen ohjausryhmä 2006–2008. Karavirta tuli ohjausryhmään Ihantolan tilalle vuoden 2008 alusta.

Nimi	Yliopisto
Petri Ihantola	Teknillinen korkeakoulu
Hannu-Matti Järvinen	Tampereen teknillinen yliopisto
Ville Karavirta	Teknillinen korkeakoulu
Ari Korhonen	Teknillinen korkeakoulu (koordinaattori)
Mikko-Jussi Laakso	Turun yliopisto
Essi Lahtinen	Tampereen teknillinen yliopisto
Lauri Malmi	Teknillinen korkeakoulu
Tapio Salakoski	Turun yliopisto

ti kaikki verkoston toiminnasta kiinnostuneet.

Tyypillisesti yksittäisen tutkijan mielenkiinto kohdistuu johonkin yksittäiseen välineeseen tai sen sisältämään menetelmään. Niinpä verkoston toimintaa on kanavoitu pilottihankkeiden muodossa, joissa välineitä on kehitetty näiden tutkijoiden yhteistyönä yli yliopistorajojen. Pilottihankkeiden tuloksia on esitelty seminaarien yhteydessä pidetyissä työpajoissa, joissa välineitä ja menetelmiä on tarjottu laajemmalle yleisölle.

Verkoston ohjausryhmässä ovat toimineet sen kolmivuotisen olemassaolon aikana taulukossa 1 mainitut henkilöt.

### 3 Tietojenkäsittelytieteiden opetuksen SIG

Tietojenkäsittelytieteiden opetuksen teemaryhmä (TKTOSIG) on osa Tietojenkäsittelytieteen Seuran toimintaa. Teemaryhmän tavoitteena on tuoda yhteen alan opettajia ja opetusteknologiasta kiinnostuneita tutkijoita. Tavoitteet ja sitä kautta toimintakin ovat siis hyvin samansuuntaiset kuin Ohjelmoinnin perusopetuksen verkostolla. Verkoston ja SIGin yhteise-

nä tarkoituksena on edistää muun muassa opettajien välistä yhteistyötä, opetus- ja oppimateriaalien vaihtoa, opetuksen tueksi kehitettyjen tieto- ja viestintätekniikkaan perustuvien välineiden käyttöönottoa sekä ottaa kantaa ajankohtaisiin yhteiskunnallisiin alan opetukseen liittyviin kysymyksiin.

Ohjelmoinnin perusopetuksen verkoston hankerahoitus on päättymässä vuoden 2008 lopussa. Toiminnan painopistettä onkin tietoisesti pyritty siirtämään perusopetuksesta koko tietojenkäsittelytieteiden opetukseen. Tästä esimerkkeinä ovat yhteistyö SIGin kanssa sekä myös sellaisten opetusvälineiden kehittäminen ja esittely, jotka eivät ole mitenkään erityisesti sidoksissa ohjelmoinnin opetukseen tai edes tietojenkäsittelytieteiden opetukseen. Toisin sanoen tavoitteena on perusopetuksessa hyvin toimivien menetelmien, käytänteiden ja työkalujen yleistäminen ja kehittäminen sellaiselle tasolle, että niitä voisivat hyödyntää muutkin kuin vain tietojenkäsittelytieteilijät — tai että niitä voisivat hyödyntää edes *kaikki* tietojenkäsittelytieteilijät. Tavoitteena on hankkeen ympärille muodostuneen verkoston toiminnan jatkumisen edellytysten turvaaminen kanavoimalla se Seuran SIG-toi-

Taulukko 2: Tietojenkäsittelytieteiden opetuksen SIGin ohjelmatoimikunta 2007–

Nimi	Yliopisto
Jonne Itkonen	Jyväskylän yliopisto
Ville Karavirta	Teknillinen korkeakoulu
Ari Korhonen	Teknillinen korkeakoulu (pj)
Mikko Laakso	Turun yliopisto
Vesa Lappalainen	Jyväskylän yliopisto
Pirjo Moen	Helsingin yliopisto
Paavo Nieminen	Jyväskylän yliopisto

minnaksi.

SIGin ja verkostohankkeen postituslistat ovat yhteisiä. Niiden kautta kotimaiset toimijat voivat välittää tietoa ajankohtaisista tapahtumista. SIGin ja verkoston jäsenet tapaavatkin usein tilaisuuksissa, joiden teemat liikkuvat tietojenkäsittelytieteiden opetuksen alueella. Kansallisesti merkittäviä tapahtumia ovat muun muassa Tietojenkäsittelytieteen päivät ja Ohjelmoinnin perusopetuksen verkoston järjestämät seminaarit ja työpajat sekä kansainvälinen Kolin kolistelu, joka on alan tutkimukseen keskittyvä tapaaminen.

SIGin ohjelmatoimikuntaan ovat kuuluneet sen perustamisesta saakka taulukossa 2 mainitut henkilöt.

## 4 Tietojenkäsittelytieteiden opetuksen kehittämisseminaari

### 4.1 Loppuraportti

Espoon Otaniemessä 30.–31. lokakuuta järjestetty Tietojenkäsittelytieteiden opetuksen kehittämisseminaari oli järjestyksessä neljäs verkoston järjestämistä seminaareista. Se oli samalla verkostohankkeen päätösseminaari, joten ohjelma alkoi hankkeen loppuraportin esittelyllä. Siinä

**Erkki Kaila** Turun yliopistosta esitteli kysely- ja haastattelututkimuksen keinoin saatuja tuloksia otsikolla *Ohjelmoinnin opetus ja opettajien suhtautuminen opetusta kehittäviin välineisiin* [3]. Työssä esitellään aluksi lyhyesti 12 lähinnä Suomessa kehitettyä välinettä, joista suurinta osaa käytetään useammassa kuin yhdessä kotimaisessa yliopistossa. Tutkimuksessa tarkastellaan näiden välineiden lisäksi ohjelmoinnin opetuksen nykytilaa ja ohjelmoinnin opettajien suhtautumista ja asenteita opetusjärjestelmiä kohtaan. Haastatteluihin osallistui 26 ohjelmoinnin ja algoritmien peruskurssien opettajaa kattaen lähes kaikki suomalaiset yliopistot, joissa ohjelmoinnin opetusta annetaan.

Tutkimus kohdistui opettajiin, kursseihin ja käytettyihin opetusjärjestelmiin. Mielenkiintoisimmat löydökset liittyivät opettajien verkottumisen ja yhteistyön määrään, kurssien samankaltaisuuksiin ja eroihin sekä opetusjärjestelmien levinneisyyteen.

Silmiinpistävää oli peruskurssien opettajien työkokemuksen vähäisyys sekä heikko verkottuminen. Yli 40% haastatelluista oli opettanut alle 10 vuotta ja vain alle 40% teki yhteistyötä muissa korkeakouluissa olevien opettajien kanssa — edes siinä määrin, että tunsu muita saman kurssin opettajia toisista yliopistois-

ta. Tämä näkyi myös siinä, että peräti 86% opettajista käytti kurssillaan itse valmistaansa luentomonistetta (tai kokoelmaa luentokalvoista).

Hieman kärjistäen voidaan todeta, että suomalainen ohjelmoinnin peruskurssi on laajuudeltaan 5 opintopistettä (68%), ohjelmointikielenä on Java (50%) ja opiskelijat saavat kurssista keskimäärin arvosanan 3 (skaalalla 0–5). Sen sijaan opiskelijamäärissä on suuria heittoja (otoksen vaihteluväli 25–760) samoin kuin keskeyttäneiden määrissä (0%–50%).

Opetusvälineitä tunnetaan kohtuullisen hyvin. Reilusti yli puolet opettajista tunsi yli puolet esitellyistä välineistä. Vähiten tunnetuimmankin välineen tunsi melkein kolmannes haastatelluista. Aika moni oli myös kokeillut välineitä käytännössä. Käyttöaste ylsi parhaimmillakin välineillä kuitenkin vain 20%:iin. Kahdestatoista välineestä vain yksi oli sellainen, jota kukaan ei enää käyttänyt (Scheme-Robo [12], jonka käyttö TKK:lla lopetettiin, kun ohjelmointikieli vaihtui Schemestä Javaan). Alkuperäisen kehittäjäyliopiston ulkopuolelle välineistä oli levinnyt setsemän. Ilmeisestikin pieni joukko opettajia adoptoi (useita) välineitä ennakkoluulottomasti ja suurin osa ei käytä mitään välinettä. Käyttöaste on josain määrin ristiriidassa sen kuvan kanssa, joka opettajilla on välineistä. Valtaosa haastatelluista oli sitä mieltä, että järjestelmät tukevat oppimisprosessia, parantavat oppimistuloksia, lisäävät opiskelumuotiivaatiota, monipuolistavat opetusta ja niin edelleen. Toki varauksiakin oli, mutta niissä korostuivat ennen kaikkea opettajan omaan työmäärään kohdistuvat paineet kuten käyttöönnoton vaikeus.

## 4.2 Kutsuttu esitelmä

Jo perinteeksi muodostunut seminaarin ulkomainen kutsuttu esitelmä sai jatkoa

tälläkin kertaa, kun esitelmöitsijöitä oli saatu peräti kaksi. **Jens Bennedsen** (IT University West, Aarhus) sekä **Michael E. Caspersen** (University of Aarhus) esitelmöivät aiheesta Programming: Creativity in Three Dimensions. Siinä tanskalaiset esittelivät kehittämänsä menetelmän nimeltä “stepwise improvement”, jossa pääpaino oli ohjelmoinnin opettamisessa on riittävien taidollisten valmiuksien luomisessa. Caspersenin mukaan menetelmä yhdistelee parhaita käytäntöjä muun muassa testauslähtöisestä ohjelmistokehityksestä sekä vallitsevista ohjelmointitavoista kuten “stepwise refinement”, jossa korostuu refaktoroinnin merkitys. Menetelmän toimivuus käytännössä oli helppo uskoa, kun esityksen aikana Bennedsen näytti miten kurssilla opettaja luennoinnin sijaan näyttää konkreettisesti miten ohjelmaa kehitellään pikku hiljaa ja muokataan paremmin vastaamaan haluttua spesifikaatiota.

## 4.3 Työpajat

Tanskalaisten mielenkiintoinen aihe keräsi myös vastaavaan työpajaan suuren joukon osallistujia. Rinnakkaisissa työpajoissa olivat kahtena päivänä esillä myös seuraavat opetusmenetelmiin ja työkalujen käytön opetteluun keskittyvät aiheet ja välineet. Lisätietoa työpajojen sisällöistä ja työkaluista löytyy verkostohankkeen kotisivuilta.

1. Motivointia Javan alkuopetukseen ja helpotusta testaamiseen, Vesa Lappalainen ja Jonne Itkonen, Jyväskylän yliopisto
2. ViLLE-tehtävät TRAKLA2:ssa, Mikko-Jussi Laakso, Teemu Rajala ja Erkki Kaila, Turun yliopisto
3. IDLE-oppimisympäristö, Markus

Makkonen ja Rami Laine, Tampereen teknillinen yliopisto

4. Rubyric, Tapio Auvinen, Teknillinen korkeakoulu
5. Invarianttiperusteinen ohjelmointi opetuksessa, Ralph-Johan Back, Johannes Eriksson ja Linda Mannila, Åbo Akademi

#### 4.4 Loppukeskustelu

Loppukeskustelussa **Tapio Salakoski** (Turun yliopisto) alusti aiheesta Ohjelmoinnin perusopetuksen rationalisointi ja konsolidaatio. Keskustelun lähtökohtana oli päättyvä hanke, jonka keskeisenä tuloksena haluttiin nähdä syntynyt verkosto. Hanke siis päättyi, mutta verkosto toivon mukaan edelleen jatkaa tulevaisuudessa seminaarien ja työpajojen järjestämistä, koska yhteistyölle tuntuu edelleen olevan tilausta.

Toinen keskeinen tema keskusteluisa oli yhteistyön laajuus. Taulukossa 3 on Salakosken jäsenitys eri yhteistyön asteista. Toisena ääripäänä on ”yksin”, jossa jokainen opettaja suunnittelee ja toteuttaa oppimateriaalit, valitsee opetusvälineet ja -muodot, asettaa oppimistavoitteet ja arvioi opiskelijoiden suoritukset täysin yksin tiedostamatta miten muut toimivat. Asteittain tietoisuus muusta maailmasta kasvaa. ”Tietoisesti” merkitsee muiden vastaavien kurssien opettajien tuntemista tai ainakin heidän kurssiensa sisältöjen tuntemista siinä määrin, että kykenee tunnistamaan erot ja yhtenäisyydet omaan kurssiinsa verrattuna. Edellinen ei kuitenkaan vielä sisällä ajatusta pyrkimykseen yhtenäistää opetusta, joka on seuraavan sarakkeen keskeinen ero tähän. Toiseen

ääripäähän siirryttäessä opetusta tehtäisiin yhteisesti tai jopa keskitetysti. Tällä hetkellä ollaan aika usein ”yksin”, mutta tavoite lienee vähintään ”tietoisesti”.

Paljon keskustelua herätti kysymys siitä, kuinka pitkälle yhteistyötä pitäisi miltäkin osin viedä. Kaikkea ei edes haluta tehdä yhteisesti tai edes yhtenäisesti vaikka siihen olisi mahdollisuksiakin. Puheenvuoroissa haluttiin toisaalta korostaa hyvien käytänteiden leviämistä ja päällekkäisen työn vähentämistä. Toisaalta haluttiin säilyttää riittävä erilaisuus ja antaa eri yksiköille mahdollisuus profiloitua. Keskustelu jatkuu seuraavassa seminaarissa, jossa on tarkoitus jatkaa jäsenyyksen ja yhteisten toimintamuotojen hakemista.

## 5 Yhteenveto

Ohjelmoinnin perusopetuksen verkosto on samannimisen verkostohankkeen seurauksena syntynyt yhteistö, jonka kotisivuilta<sup>2</sup> löytyy lisätietoa muun muassa sen järjestämistä seminaareista ja työpajoista. Sivut ovat osittain yhteiset Tietojenkäsittelytieteiden opetuksen SIGin kotisivujen<sup>3</sup> kanssa. Muun muassa linkkilista erillisiin opetustyökaluihin sekä postituslistasta (opeverkosto@cs.hut.fi) ja sen arkisto ovat yhteisiä.

Verkoston sivuilta löytyy lisätietoa myös pilottihankkeista (13 kpl), joissa välineitä on pyritty siirtämään alkuperäisistä kehittäjäyliopistoista muihin yliopistoihin. Pilottihankkeissa työstettyjä välineitä ja materiaaleja on esitelty verkoston järjestämässä seminaareissa ja työpajoissa. Työpajojen välineet ja aineistot on julkaistu ja jaettu osallistujille myös USB-tikulla, jonka sisältö löytyy myös verkosta<sup>4</sup>. Jokaisesta työpajoissa esitellyistä välinees-

<sup>2</sup><http://www.cs.hut.fi/Research/COMPSER/Verkosto/>

<sup>3</sup><http://www.cs.hut.fi/Research/COMPSER/TKTOSIG/>

<sup>4</sup><http://verkkoopetus.cs.utu.fi/vhanke/Aloita.html>



Taulukko 3: Opetusyhteistyön osa-alueet ja laajuus. Tapio Salakosken jäsenyys.

	yksin	tietoisesti	yhtenäisesti	yhteisesti	keskitetysti
materiaalit					
välineet					
opetusmuodot					
oppimistavoitteet					
arviointi					

tä löytyy vähintään kuva ja lyhyt tiivistelmä sekä mahdollisesti myös esittelyvideo, pidempi artikkeli, kotisivun osoite ja muuta lisätietoa.

Verkostolla on myös oma Wiki<sup>5</sup>, jonne voidaan jatkossa kerätä välineitä, työkaluja, oppimateriaaleja ja muuta tietoa verkoston toiminnan seurauksena syntyneistä parhaista käytännöistä.

## Kiitokset

Kiitokset Antti Valmarille ja Jorma Tarhiolle tämän tekstin aikaisempien versioiden huolellisesta oikoluvusta.

## Viitteet

- [1] T. Ahoniemi ja T. Reinikainen. Aloha — a grading tool for semi-automatic assessment of mass programming. Anders Berglund & Mattias Wiggberg (toim.): *Sixth Baltic Sea Conference on Computing Education Research (Koli Calling 2006)*, sivut 139–140, 2006.
- [2] Formulaattori. <http://formulaattori.cs.tut.fi/>, [viitattu 10.12.2008].
- [3] Erkki Kaila. Ohjelmoinnin perusopetuksen verkostohankkeen loppuraportti. Kandidaatintutkielma, Turun yliopisto, valmistuu vuoden 2009 aikana.
- [4] Ari Korhonen. Visuaalinen algoritmimulaatio ja sen sovelluksia. *Tietojenkäsittelytiede*, (23):42–59, 2005.
- [5] Mikko-Jussi Laakso, Tapio Salakoski, Linda Grandell, Xuemei Qiu, Ari Korhonen ja Lauri Malmi. Multi-perspective study of novice learners adopting the visual algorithm simulation exercise system TRAKLA2. *Informatics in Education*, 4(1):49–68, 2005.
- [6] Mikko-Jussi Laakso, Tapio Salakoski ja Ari Korhonen. The feasibility of automatic assessment and feedback. *Proceedings of Cognition and Exploratory Learning in Digital Age (CELD A 2005)*, sivut 113–122, Porto, Portugali, joulukuu 2005. IEEE Technical Committee on Learning Technology and Japanese Society of Information and Systems in Education.
- [7] Mikko-Jussi Laakso, Tapio Salakoski, Ari Korhonen ja Lauri Malmi. Automatic assessment of exercises for algorithms and data structures — a case study with TRAKLA2. Ari Korhonen & Lauri Malmi (toim.): *Fourth Finnish/Baltic Sea Conference on Computer Science Education (Koli Calling 2004)*, sivut 28–36, 2004. Teknillinen korkeakoulu.
- [8] Lauri Malmi, Ville Karavirta, Ari Korhonen, Jussi Nikander, Otto Seppälä ja Panu Silvasti. Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. *Informatics in Education*, 3(2):267–288, 2004.

<sup>5</sup><https://wiki.tkk.fi/display/opeverkosto/Ohjelmoinnin+perusopetuksen+verkosto>

- [9] Niko Myller, Mikko Laakso ja Ari Korhonen. Analyzing engagement taxonomy in collaborative algorithm visualization. *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, sivut 251–255, Dundee, Skotlanti, 2007. ACM.
- [10] Teemu Rajala, Mikko-Jussi Laakso, Erkki Kaila ja Tapio Salakoski. Language-independent program visualization tool. Raymond Lister (toim.): *Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007)*, sivut 151–159, 2007. ACS.
- [11] Rubyric. <http://svg.cs.hut.fi/Rubyric/>, [viitattu 10.12.2008].
- [12] Riku Saikkonen, Lauri Malmi ja Ari Korhonen. Fully automatic assessment of programming exercises. *Proceedings of The 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*, sivut 133–136, Canterbury, UK, 2001. ACM.